Trojan Horse in DRE -- OS

Taxonomy

Configuration-related, Change Management, COTS, OS, Trojan.

Applicability

DRE, DRE with VVPT

Method

A third party supplies a well known, publicly available operating system used in a DRE. The attack team introduces a Trojan horse that is activated on a specific date (e.g., the first Tuesday after the first Monday in November). The Trojan horse detects when a ballot is displayed, and reverses the order of the first two entries on the screen (so if the order should be, for example, John Adams and Tom Jefferson, the displayed order is Tom Jefferson and John Adams). The Trojan horse also checks for the names on the review screen and if either name appears, the other is substituted.

If desired, the trigger can be some event other than a date -- for example, if a voter selects and then cancels a certain candidate four times in a row, or if three voters sequentially vote for the selected candidate, then those patterns can be a trigger for the Trojan horse.

The Trojan horse is inserted when a piece of the operating system is rewritten, either by the perpetrator or by someone whom the perpetrator has compromised (bribed, blackmailed, etc.) The driver is NOT written by the DRE software developer, and hence is COTS software. The Trojan horse code can be placed within any OS components that are known to be configurable for a running installation, such as the video driver, the user interface devices, the drivers for removable storage, etc., that would not be part of the standard COTS as delivered by the vendor, but would be expected to test positive for signs of change.

Resource Requirements

Access to each voting machine, the host machine(s) on which the master images of the OS or application software are created and/or stored, or any intermediary system(s) that might be responsible for installing software onto the voting machines. This would also include access to those systems responsible for delivering software patches or updates to the DRE over the course of the system's operating lifetime.

Potential Gain

Most operating system functions are executed within privileged space in the system architecture, which means that they have both the rights and the ability to make any and all changes they wish to any part of the system, including those routines (such as audit logs) that are supposed to detect inappropriate behavior. As such, a successful attack on the DRE OS would open the door to any tampering that the attack team could create.

Likelihood of Detection

Detection would depend upon the (TBD) rigor of the Voting System Testing Laboratory (VSTL) examination process, and/or the pre-election testing of the voting system. Testing would have to be conducted in such a fashion that the complete ballot input and output datasets would be validated, and that the system would not offer evidence to the trojan horse code, such as entering a defined "test mode", that would enable the code to mask its presence and remain dormant throughout the testing.

Countermeasures

This attack presents little in the way of a knowable profile, making countermeasures almost impossible. Fingerprinting of the OS in the form of hashing an approved version would capture a trojan horse in the core functions, but would not include those modules and drivers that are typically reconfigured upon installation of the OS on a given device (or set of devices).